Indonesian Journal of Electrical Engineering and Computer Science
Vol. 17, No. 1, January 2020, pp. 389~397
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v17.i1.pp389-397

A light weight encryption over big data in information stockpiling on cloud

Uma Narayanan[1], Varghese Paul[2], Shelbi Joseph[3]
[1,3]Division of Information Technology, Cochin University of Science and Technology, India
[2]Department of Computer Science and Engineering, Rajagiri School of Engineering and Technology, India

Article Info

Article history:

Received Apr 20, 2019
Revised Jul 22, 2019
Accepted Aug 6, 2019

Keywords:

AES
HDFS
Ranger key management
SALSA20
Security

ABSTRACT

Data is growing exponentially in the fast Changing World of Information and Communications Technology. Information from sensors, cell phones, social networking sites, logical information and ventures all are adding to this gigantic blast in the information. One of the best mainstream utilities available for dealing with the colossal measure of data is the Hadoop community. Enterprises are progressively depending on Hadoop for preparing their essential information. In any case, Hadoop is still developing. There is much powerlessness found in Hadoop, which can scrutinize the security of the sensitive data that undertakings have entrusted on it. In this paper, security issues related to the system have been discussed. Besides, we have attempted to give a short overview of the currently accessible arrangements and their constraints. Towards the end, a novel strategy, which can be utilized to kill the detected vulnerabilities in the structure, has been introduced. In the cutting edge period, data security has moved towards becoming a basic need for every single person. Be that as it may, not every person can bear the cost of the specific circulations given by various sellers to their Hadoop group. This paper displays an effective system that anybody can use with their Hadoop to secure Data.

Corresponding Author:

Uma Narayanan,
Division of Information Technology,
Cochin University of Science and Technology, Kerala, India.
Email: uma@cusat.ac.in

1. INTRODUCTION

A large volume of data is being generated every day. Data from Sensors, Mobile phones, Transaction data, Social media Enterprise data, and Public data are adding to this torrent of Data. This sudden deluge can be understood by how we have made a larger volume of data over the latest five years. Big Data as these large bits of information is all around has ended up being a standout amongst the most sweltering investigation drifts today. Enormous Data is the aggregation of the mass volume of information, and that information can be in any shap as an organized frame or unstructured frame. It is broadly well known in a few fields because of its stockpiling limit of social and non-social, organized and unstructured data. For enormous associations and business advancements, it is a chance to upgrade business. The forecast for revenue big data in America (in billion U.S. dollar) itself is shown in Figure 1, which shows the importance of big data in the future.

Data is delivered in sweeping aggregate in the light of correspondence and transmission of data, and the Big Data ought to have been set up for information mining computations. The need is to create proficient frameworks that can use this possibility to the most extreme, remembering the present difficulties related to its structure, scale, security and its analysis. There has been a move in the engineering of information preparing frameworks today, from the centralized architecture to the distributed architecture. For our research work, we are utilizing mobile data. There are vast amounts of data available in Mobile, so we are using the

data in cloud storage for further analysis. Since we are dealing with a massive amount of data for analysis it is stored in HDFS. Security is one of the basic features to keep information shielded and secure from unfortunate and unintended data. Examination of existing work reasons that HDFS does not have any security structure or count to keep information shielded and secure. This work proposes a response for performing encryption of Big Data going to be put into HDFS as ensured and secure. The Hadoop Distributed File System (HDFS) [2] is a distributed file system. HDFS is developed as low-cost hardware, which is highly fault-tolerant. HDFS provides us with high throughput when the large data set is used, so it was considered more suitable for application that makes use of Big Data. A typical file in HDFS is gigabytes to terabytes in size. Thus, HDFS [3] is tuned to support large files. It should provide high aggregate data bandwidth and scale to hundreds of nodes in a single cluster. It should support tens of millions of files in a single instance.

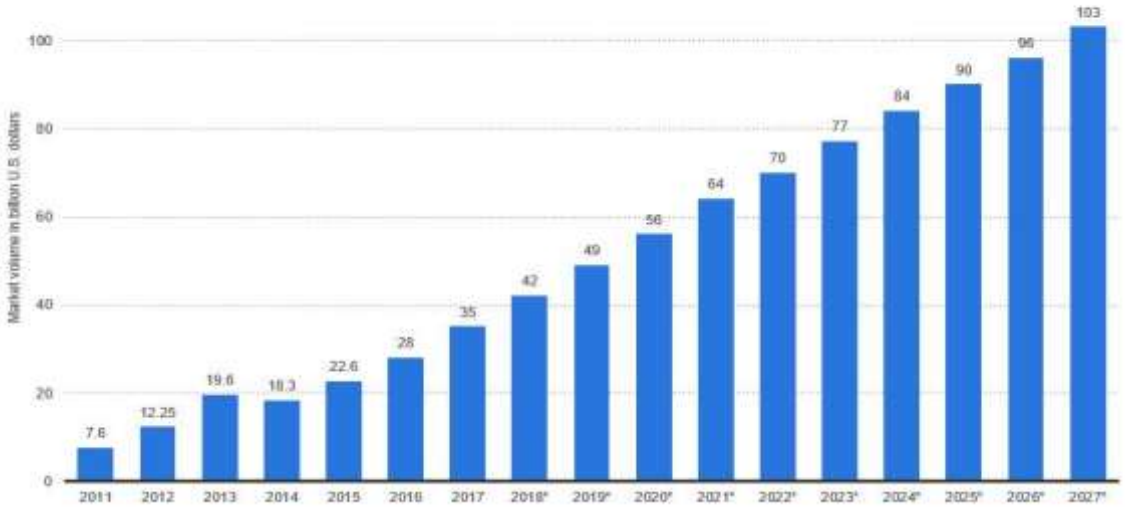

Figure 1. Forecast of Big Data market size, based on revenue from 2011 to 2027 [1]

Big Data research continually encounters Hadoop. Hadoop is expected to process an immense proportion of data, paying little mind to its structure. The focal point of Hadoop is a MapReduce framework, made by Google to deal with the issue of web search indexes. The non-benefit association Apache Software Foundation (ACF) maintains and manages the Hadoop framework and Hadoop environment technology [4] and [5]. The Map-Reduce [6] is a programming model of Hadoop. Security systems are not incorporated at HADOOP [2]. A few works have been accounted for the utilization of cryptographic calculations to scramble the data and store the data at HDFS. Encryption is utilized to give security to sensitive data. Encryption calculation performs different substitutions and changes on the first message or information and changes it into ciphertext, which at last turns into an arbitrary message. There are a couple of various developments made in the Hadoop condition to play with Enormous Information and ace one's specific capacities. The MapReduce framework has been comprehensively grasped by multiple associations and relationship to process the epic volume of datasets along these lines; it deals with the issue of data that is guideline speaking excessively far-reaching.

Big data is gigantic and untidy, and it's coming to you uncontrolled. Data is accumulated to be dissected to find examples and relationships that couldn't be evident at first, yet may be valuable in settling on business choices in an association. Security and protection of Big data are the most significant tests since they dependably dwell into sharable capacity put known as HDFS. Along these lines, the fundamental need for the advancement of Big Data examination device is to defeat these difficulties and recover data with an ideal solution. The data is lost because of sensitivity, to diminish the data drop the data are scrambled before stacking the data into the cloud. The transformation of plain content to ciphertext is called encryption, and the turnaround procedure is called decryption. Numerous encryption calculations are broadly accessible and utilized in data security. They can be sorted into Symmetric (private) and Asymmetric (public) keys

encryption. In Symmetric keys encryption or secret key encryption, just a single key is utilized to scramble and decode information. In Asymmetric keys, two keys are utilized: private and public keys. A study of different symmetric and symmetric cryptography methods with existing vulnerabilities, countermeasures and comparison was done [7]. Bih-Hwang Lee et al. [8] discuss data security in cloud computing using AES under the Heroku cloud. They implement Heroku cloud (example of a cloud platform as a service) then implemented AES in the website to secure data. Table 1 shows the comparison of AES, DES, and RSA.

Table 1. Comparison between AES, DES and RSA [9]

Factors	AES	DES	RSA
Key Size	128, 192, 256 bits	56	>1024
Ciphering & deciphering key	Same	Same	Different
Algorithm	Symmetric Algorithm	Symmetric Algorithm	Asymmetric Algorithm
Encryption and Decryption	Faster	Moderate	Slower
Power Consumption	Low	Low	High
Security	Excellent Secured	Not Secure Enough	Least Secure
Hardware & Software Implementation	Faster	Better in hardware than in software	Not Efficient
Ciphering & Deciphering	Different	Different	Same

Lin, H.Y., et al. [10] tended to the information privacy issue by coordinating half and half encryption plans and the Hadoop distributed file system (HDFS). They proposed two mixes, HDFS-RSA and HDFS-Pairing, as augmentations of HDFS. They have presented incorporation of crossover encryption plans and HDFS as an option secure capacity framework for Hadoop. It is expressed that HDFSRSA and HDFS-Pairing have significant overhead on composing tasks and adequate overhead on perusing activities and HDFS-RSA and HDFS-Pairing are suitable for write-once read-many applications. Cohen and Acharya [11] proposed an AES based New Instruction (AESNI) encryption structure for information encryption and uprightness approval by making utilization of the Trusted Platform Module (TPM). Security and privacy are some of the most critical issues of cloud data services. A survey of several state-of-the-art security solutions has been proposed for protecting outsourced data, and user privacy is mentioned [12].

Privacy as a Service is used to protect the storage and handling of users' private data [13]. Cloud computing has utilized in marketing field because of the consistency, wage for each usage, fault-tolerance, and scalability [14]. Recommended the utilization of symmetric encryption algorithm to be received in IoT, they [15] proposed execution of the Blowfish encryption algorithm on FPGA asset and programmed it utilizing the VHDL language. The proposed usage was assessed by estimating execution measurements, for example, security, encryption time, avalanche impact, and throughput and found to give a decent exhibition. Current security practices are required to guarantee the availability, integrity, privacy, and confidentiality of outsourced data. The paper [16] highlights the leading security challenges of the cloud storage service and introduces some solution to address those challenges.

The big data security utilizing data masking technique is discussed in the paper [17]. A lot of efforts have been put by various specialists to make it basic, simple, powerful and efficient utilization of big data. In this overview paper [18] accentuated on the working of Map Reduce, challenges, opportunities with the goal that analysts can think on further improvement. In the paper of Hamid Bagheri et al., [19] three points are recommended for research heading: Security issues in Big Data, context-aware information retrieval, and integrating ontology with Big Data. In paper [20] an outline of big data is presented along with Big data usages and several challenges that are associated with big data. The issues identified with the clustering methods in data mining. Sachin Arun Thanekar et al. [21] feature the upsides of Hadoop in handling the high volumes of data through correlation of Hadoop and RDBMS. The work [22] concentrated on the issue of security of the variety of data that is available on a major platform and planned to give a methodology that could ensure sensitive information. To tending to this objective, proposed a novel methodology entitled as split and merge strategy to assess the normal execution measurements, assessed the model by evaluating its execution time and size while diverse input sizes were worked.

Bhargavi I. et al., [23] states that ordinarily distributed storage is utilized for putting away enormous information. In spite of viable cost sparing, distributed storage is inclined to numerous security dangers. The privacy of the clients' information is a basic issue at cloud-based administrations. At a base dimension of cloud-based capacity, information security is a basic issue and inclined to security violations. The primary security violations are Data Leakage, Unauthorized access, Denial of service of Resources. The three key components on verifying Big Data on General cloud-based capacity are given as Integrity, secrecy, and accessibility. To do near the investigation, they have thought about ECC, RSA, AES, DES, and Elliptic bend Diffie-Hellman to confirm the BIG Data security at a cloud-based data center. The paper [24] made a thorough assessment of cryptographic algorithm DES, 3DES, AES, RSA and Blowfish and landed at the end

dependent on their execution results got on the assessment parameters such as encryption time, decryption time, memory utilized and so on., that AES can be utilized in applications where secrecy and uprightness is of most noteworthy need. Also, Blowfish is strongest against guessing attacks.

Hadoop is changing the impression of dealing with Big Data particularly the unstructured information. To see a fundamental idea concerning Hadoop security, one needs to look back at the base of Hadoop. Security was not the basic requirement for Doug Cutting and his gathering at the time they started developing the Hadoop [25]. To begin with, it was just some part of Apache Nutch wander until the point that it was moved to the new Hadoop subproject in January 2006. Study of past research work and Big Data applications create an issue of instability and protection spillage issue. We have recognized three classes of security infringement: the unapproved arrival of data, unapproved alteration of data and disavowal of resources [26]. Data Nodes forced no entrance control; an unapproved client could read personal information hinders from Data Nodes, bypassing access control component/limitations, or composing refuse information to Data Node [27]. Since we are storing the data into the cloud using the mobile application, we need an efficient and fast algorithm. Here we identify the most suitable algorithm that is most appropriate for our work.

The primary objective of this paper is to devise a technique by which data should be securely sent and recovered in a big data environment and the meantime it should not increase any overheads in computation, storage or communication. The procedure utilized as a part of the proposed work depends on a legitimate framework. This work proposes supplanting of the AES algorithm with the Salsa20 algorithm. Information encryption is one of the fundamental necessities in the Hadoop framework to keep data private and safe from unapproved use. Encryption algorithm dependably accompanies the issue of additional overhead which should endeavor to diminish as conceivable. AES algorithm in existing work can be supplanted by Salsa20 to enhance the execution of Hadoop ecosystem during data encryption. In this way, key size can likewise be raised from 128 bits to 256 bits for symmetric key cryptography. For a unified security system to oversee fine-grained get to control crosswise over Hadoop parts we utilize Apache Ranger. The proposed course of action would perform Salsa20 encryption first before moving into HDFS and Information Mining computation with Mapper class to perform parallel getting ready of encryption alongside mining on ciphered information.

2. RESEARCH METHOD

The Salsa20 [28] is stream cipher expands a 256-bit key into 264 randomly accessible streams, each containing 264 randomly accessible 64-byte blocks. The 64-byte input x to Salsa20 is viewed in little-endian form as 16 words x0, x1, x2, ..., x15 in {0,1,...,2^32-1}. These 16 words are fed through 320 invertible modifications, where each modification changes one word. The resulting 16 words are added to the original x0, x1, x2, ..., x15 respectively modulo 2^32, producing, in little-endian form, the 64-byte output Salsa20(x). Each modification involves xor'ing into one word a rotated version of the sum of two other words modulo 2^32. Thus the 320 modifications involve, overall, 320 additions, 320 xor's, and 320 rotations. The rotations are all by constant distances. The entire series of modifications is a series of 10 identical double-rounds. Each double-round is a series of 2 rounds. Each round is a set of 4 parallel quarter-rounds. Each quarter-round modifies four words [28].

a. There is the mathematical function of the quarter round.
b. Then row round function, this round will modify the rows of the matrix.
c. After that, there is the column round function where the round will modify the column in the matrix.
d. Then proceed with a double round function which the function will do looping as many as have been specified, 20 rounds for salsa20/20 or 8 rounds for salsa20/8.
e. After that, there is a little-endian function.

Based on the mathematical function, we can get the 64byte output as a keystream. Encryption, authentication, and platform management tools are significantly enhancing the security of Hadoop clusters, deterring all the most effortless ways aggressors have used to take data or trade-off usefulness. Encryption ensures two strategies for going around application security controls. Furthermore, encryption is obligatory if you have to fulfill consistency or information administration necessities. In our unique research we stressed most on consolidating ten Hadoop modules, all sent with specially appointed configurations, and clouded inside the complexities of the cluster, each exposing its unique attack surface to adversaries. Deployment validation stays at the highest priority on our rundown of concerns; however, Apache Ranger gives a steady administration plane to setting up configurations and uses strategies to protect data within the cluster. The Ranger Key Management Service (Ranger KMS) provides a scalable cryptographic key management service for HDFS “data at rest” encryption. Ranger KMS depends on the Hadoop KMS initially created by the Apache people group and expands the local Hadoop KMS usefulness by permitting framework

administrators to store enters in a safe database. Utilizing the Apache Ranger console, security executives can without much of a stretch oversee approaches for access to records, organizers, databases, tables, or segment. Apache Officer has fine-grained approval to complete a particular activity or potentially task with Hadoop component/tool and oversaw through a central administration tool. Standardize the approval technique overall Hadoop segments.

3. RESULTS AND ANALYSIS

Experimental setup, a 10-node Hadoop cluster was configured. The first eight nodes provided both computation (as MapReduce clients) and storage resources (as DataNode servers), and the 9th and 10th node served as both the MapReduce scheduler and NameNode storage manager. First, eight nodes was an 8-V processor running at 8GB of RAM and a gigabit Ethernet NIC. All nodes used Hadoop framework 2.7.7, and Java 1.7.0. The last two-node 16-V processor and, 32GB of RAM. Disk stored the operating system, Hadoop application, and application scratch space, stored only HDFS data. Hadoop replication was disabled. The sequence stages in the process of encryption and decryption of Mobile data packets in a cluster of 10 nodes application push to store data at HDFS. Various cryptographic algorithms are available and were used in information security. There are different types of algorithms: (i) Symmetric-key algorithms such as Data Encryption Standard (DES), Triple DES and Advanced Encryption Standard (AES) (ii) Asymmetric-key algorithms such as RSA and Elliptic Curve Diffie-Hellman (ECDH). For comparison purposes, we have considered AES and Blowfish approaches since they are widely adopted in current works [29-33]. So the comparison of AES and BlowFish algorithm with our proposed approach is made on different data size varying from 1MB to 10 GB data. Encryption time for different algorithms as shown in Figure 2.

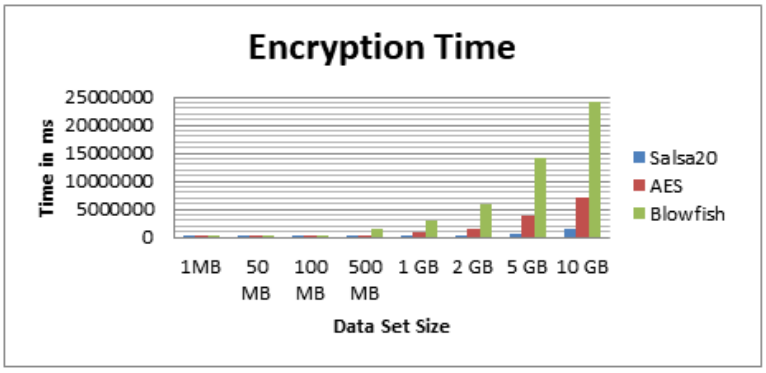

Figure 2. Encryption time for different algorithms

Every one of the encryption methods has its very own strength and weakness. To apply a reasonable cryptography calculation to our application, we should have information in regards to execution, quality, and shortcoming of the different algorithms. In this paper, we have investigated the following measurements under which the cryptosystems can be most suitable for our work. They are, first one is Encryption time estimated in milliseconds. Encryption time impacts the execution of the framework. Encryption time must be less making the framework quick and responsive. The second parameter is Decryption time which is to be less like encryption time to make framework responsive and quick. In our test, we have estimated decryption time also in milliseconds. Third one and also most important parameter is Avalanche effect: A small change in input will result in a significant difference in the output message is called Avalanche effect. A high impact of avalanche is most desirable for a good algorithm. So the three parameters and the impact on our experiment in details are shown below. The time taken by the different algorithm for varying data size is shown in Figure 2. The analysis result shows that our proposed algorithm takes lesser time for encryption of even larger data size. Result suggests that Salsa20 is the most suitable for our application. Salsa20 is designed with the end that it can encode any arbitrary block of data, making it appropriate to be utilized as a block cipher. Another significantly preferred standpoint of this technique is that each encrypted block can be independently decrypted given you have the key and nonce for the decoding. Time taken for decoding appears in Figure 3. Consequently, MapReduce tasks can be explicitly executed on encrypted data put away in HDFS.

The throughput of decryption of AES is greater than encryption. However, there are a few reasons why it seems distinctive while encrypting/decrypting multiple blocks. For instance, with cipher-block chaining (CBC), encryption must be done successively (encode block 0 preceding you can encrypt block one preceding you can encrypt block two ...), while decryption can be parallelized as the XOR step (with the previous block of ciphertext) is done after the block cipher is applied. The throughput of the encryption/ decryption scheme is calculated by dividing the total plaintexts in Megabytes over the total encryption/decryption time (millisecond) [34] as in (1). So, as the throughput of an algorithm increased, our algorithm is considered to have a high speed.

The throughput of the algorithm given in Figure 4, from here it's clear that Salsa20 take lesser time for both encryption and decryption surely indicates that our suggested Salsa20 is the most appropriate for our research work.

Throughput= Plain text (Megabyte) / Encryption Time (Millisecond). (1)

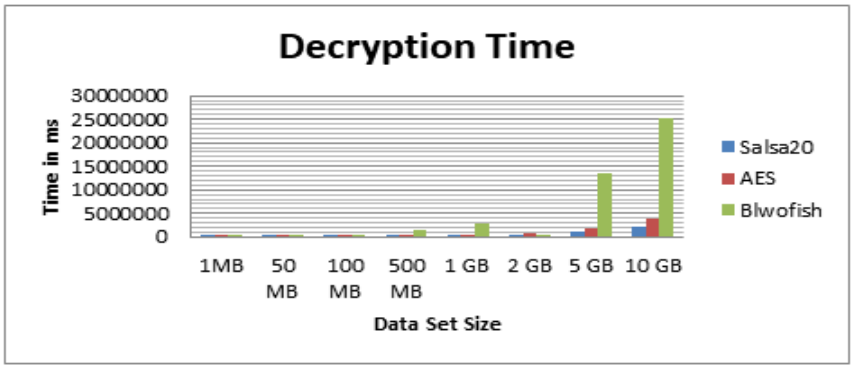

Figure 3. Decryption time for different algorithms

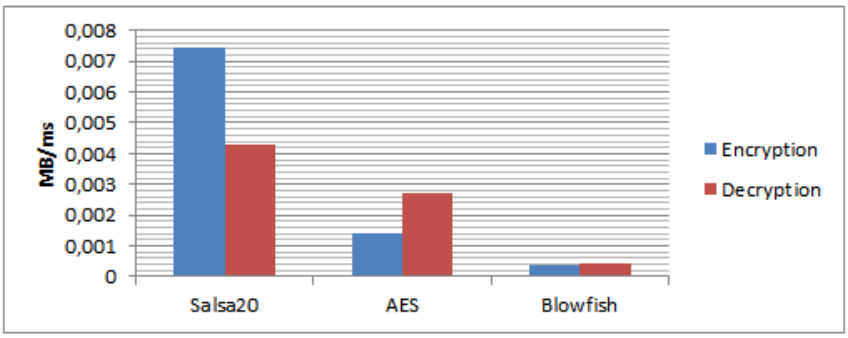

Figure 4. Performance analysis of algorithms

Figure 4 shows the superiority of the Salsa20 encryption algorithm over all other algorithms for all intervals, from small mobile file sizes to big mobile file sizes. Result indicates that Salsa20 has high speed for encrypting mobile big data per second in comparison with the other algorithms. Another point can be noticed that Blowfish has a very low throughput, due to the small key size used when compared with the other algorithms. In Security investigation, we have assessed our proposed algorithm by the security metric called the Avalanche Effect. Note that the key advantage of Avalanche Effect is to quantify the quality of the proposed calculation against breaking and hacking dangers and continuous assaults, for example, brute force assaults. In key encryption algorithms, changing a few bits in the original message to avalanche changed in coming round bringing about countless content piece changes. For an algorithm to fulfill the avalanche criterion, the difference in one plaintext bit is relied upon to result in one-half of the ciphertext [35]. The avalanche effect impact can be determined as in (2):

Avalanche effect = (Numbe of flipping bits in the ciphertext / Number of bits in the ciphertext) *100% (2)

The outcomes in Table 2 [36] are in the wake of figuring the individual Avalanche Effects. Figure 5 exhibits the avalanche impact on Salsa20 and other benchmark encryption calculations. It might be seen that AES and Salsa20 are the primary encryption to satisfy the avalanche impact standard; as such, it needs a greater chance to be broken. That is the reason Salsa20 can be used and is extensively more reliable than other benchmark algorithms.

From the evaluations, it's concluded that the proposed framework with Salsa20 has the lowest running time, highest throughput, and highest avalanche effect. Thus, the proposed novel framework is fast, efficient, secure, reliable, and scalable. Nothing suggested here harms performance, scalability, or functionality. We hope you find this research helpful.

Table 2. Avalanche Effect for Different Encryption Algorithms

Encryption Technique	Number of bits flipped	Percentage
DES	29	24.2
3-DES	37	30.8
MARS	59	49.2
Blowfish	31	48.4
AES	63	52.5
Salsa20	71	56.6

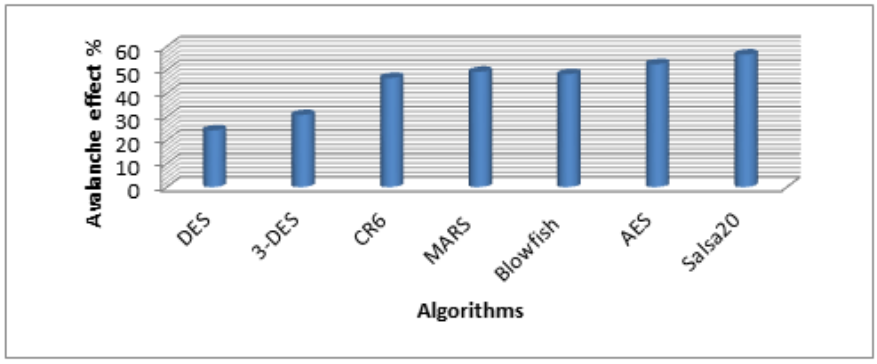

Figure 5. Avalanche effect the percentage of different algorithms

4. CONCLUSION

The total work infers that there is an urgent need to give information encryption arrangement over plain content in the Hadoop system. The proposed method would utilize Salsa20 algorithm for encryption purpose and Ranger Key Management Service (Ranger KMS). The total work will be assessed on the premise of transmission speed and calculation time for a multi-node Hadoop cluster. The complete work concludes that proposed solution ensures confidentiality, authentication and access control on the Hadoop server. It assesses the client with rights and access authorization before mining. This procedure guarantees not just approval of access; rather it additionally filters the unwanted and extra mining effort along with privacy layer between actual owner data and user. Ten nodes Hadoop cluster gives low computation time. So proposed novel approach shows that this arrangement can be utilized with Hadoop to keep up security in HDFS for substantial information especially for the storage of mobile information on the cloud which provides security with minimal computation time. This paper distinguishes the qualities, restrictions, and current research trends in Big Data Security in the cloud environment. This learning is vital for Big Data researchers. They could use the discoveries of this examination to devise altered information security models. It likewise encourages the business to comprehend the operational effectiveness of content mining methods. It further adds to diminishing the expense of the undertakings and supports compelling basic for Big data decision making. Practitioners, specifically data analysts, should consider the results highlighted in the findings and adopt such recommendations when storing data in the cloud. The findings point out, a fast, efficient, secure, reliable, and scalable algorithm for storing Big data in the cloud, having knowledge of this, and using the right technology to store the message will reduce the overall cost.

As a piece of future work, the created framework will be kept running on other Big data application, such as IoT in a military application , Big data in education, IoT for HealthCare Business.

ACKNOWLEDGEMENTS

I respect and thank Prof. Dr.Varghese Paul, for providing me an opportunity to do the project work in CUSAT and giving me all support and guidance which made me complete the project. I am extremely thankful for providing such a nice support and guidance, although he had busy schedule.

REFERENCES

[1] Big Data market revenue forecast worldwide 2011-2027", by Wikibon; SiliconANGLE Available: https://www.statista.com/statistics/254266/global-big-data-market-forecast/. Web. February.04 2019.
[2] Apache Hadoop 3.2.0-Hdfs Architecture, Available: https://hadoop.apache.org/docs/r3.2.0/hadoop-project-dist/hadoop-hdfs/HdfsDesign (accessed January 04, 2019).
[3] D. Borthakur, ''The Hadoop distributed file system: Architecture and design,'' *Hadoop Project Website*, vol. 11, pp. 21, Aug. 2007.
[4] Meenal Dhatrak, Himanshu Panadiwal,"Privacy-Preserving Mining using Data Encryption scheme for Hadoop Ecosystem," in *International Journal of Advanced Research in Science, Engineering and Technology* Vol. 5, Issue 4, April. 2018.
[5] Shalini Singh, Meena Sharma, "The Prototype for Implementation of Security Issue in Big Data Application using Hadoop Server," *International Journal of Computer Applications* (0975-8887) Volume 145-No.13, July 2016.
[6] S. Ghemawat and J. Dean, ''MapReduce: Simplified data processing on large clusters,'' *ACM Commun. Mag.*, vol. 51, no. 1, pp. 107-113, Jan. 2008.
[7] Yogesh Kumar, Rajiv Munjal, and Harsh, "Comparison of symmetric and asymmetric cryptography with existing vulnerabilities and countermeasures", *(IJAFRC)* Volume 1, Issue 6, June 2014. ISSN 2348-4853.
[8] Bih-Hwang Lee and Ervin Kusuma Dewi, and Muhammad Farid Wajdi (2018) "*Data security in cloud computing using AES under HEROKU cloud*" in The 27th Wireless and Optical Communications Conference (WOCC2018) pp. 1-5.
[9] Prerna Mahajan & Abhishek Sachdeva, " A Study of Encryption Algorithms AES, DES, and RSA for Security", *Global Journal of Computer Science and Technology Network, Web & Security*, Vol.13, Iss. 15, Vol. 1. 2013.
[10] Hsiao-Ying Lin, Shiuan-Tzuo Shen, Wen-Guey Tzeng, Bao-Shuh P Lin, "*Toward data confidentiality via integrating hybrid encryption schemes and Hadoop Distributed File System*," in the Proceedings of the 2012 IEEE 26th International Conference on Advanced Information Networking and Applications (AINA), pp. 740-747. 2012.
[11] J. Cohen, S. Acharya (2013), "*Towards a Trusted Hadoop Storage Platform: Design Considerations of an AES Based Encryption Scheme with TPM Rooted Key Protections*," IEEE 10th International Conference on and Autonomic and Trusted Computing (UIC/ATC), Ubiquitous Intelligence and Computing, pp. 444-451.
[12] J. Tang, Y. Cui, Q. Li, K. Ren, J. Liu, and R. Buyya, "Ensuring security and privacy preservation for cloud data services," *ACM Computing Surveys* vol. 49, no. 1, pp.13. 2016.
[13] W. Song, B. Wang, Q. Wang, Z. Peng, W. Lou, and Y. Cui, "A privacy preserved full-text retrieval algorithm over encrypted data for cloud storage applications," *Journal of Parallel and Distributed Computing*, vol. 99, pp.14-27. 2017.
[14] P. Vijaya Bharati, and T. Sita Mahalakshmi, "Data storage security in cloud using a functional encryption algorithm," In *Emerging Research in Computing, Information, Communication and Applications*, Springer Singapore, pp. 201-212. 2016.
[15] Prasetyo K N, Purwanto Y, Darlis D (2014)"*An Implementation of data encryption for internet of things using blowfish algorithm*." on FPGA. International Conference on Information and Communication Technology.
[16] A. Abo-alian, N. L. Badr, and M. F. Tolba, "Data Storage Security Service in Cloud Computing: Challenges and Solutions," In *Multimedia Forensics and Security, Springer International Publishing*, pp. 25-57.2017.
[17] Archana RA, Ravindra S Hegadi, Manjunath TN. "A Big Data Security using Data Masking Methods". *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*. 2017; 7(2):449-456. DOI: 10.11591/ijeecs.v7.i2.pp449-456
[18] Sachin Arun Thanekar, K. Subrahmanyam, A.B. Bagwan. "Big Data and MapReduce Challenges, Opportunities and Trends", *International Journal of Electrical and Computer Engineering (IJECE)* Vol. 6, No. 6, December 2016, pp. 2911-2919 ISSN: 2088-8708, DOI: 10.11591/ijece.v6i6.10555
[19] Hamid Bagheri, Abdusalam Abdullah Shaltooki "Big Data: Challenges, Opportunities and Cloud Based Solutions", *International Journal of Electrical and Computer Engineering (IJECE)* Vol. 5, No. 2, April 2015, pp. 340-343 ISSN: 2088-8708
[20] Chetna Kaushal, Deepika Koundal, "Recent trends in big data using hadoop." *International Journal of Informatics and Communication Technology (IJ-ICT)* Vol.8, No.1, April 2019, pp. 39~49 ISSN: 2252-8776, DOI: 10.11591/ijict.v8i1.pp39-49
[21] Sachin Arun Thanekar, K. Subrahmanyam, A.B. Bagwan. "A Study on MapReduce: Challenges and Trends". *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*. 2016; 4(1):176-183. DOI: 10.11591/ijeecs.v4.i1.pp176-183.
[22] Salisu Musa Borodo, Siti Mariyam Shamsuddin, Shafaatunnur Hasan, "Big Data Platforms and Techniques", *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*Vol. 1, No. 1, January 2016, pp. 191-200 DOI: 10.11591/ijeecs.v1.i1.pp191-200

[23] Bhargavi I., Veeraiah D., Maruthi Padmaja T., "Securing BIG DATA: A Comparative Study Across RSA, AES, DES, EC and ECDH". In *Computer Communication, Networking and Internet Security*. Lecture Notes in Networks and Systems, vol 5. pp 355-362, Springer, Singapore.2017.
[24] Priyadarshini Patila, Prashant Narayankar, Narayan D G, Meena S M (2015), "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish", in *International Conference on Information Security & Privacy (ICISP2015)*, 11-12 December 2015, Nagpur, INDIA.
[25] B. Lakhe, Practical Hadoop Security. New York, NY, USA: Apress, pp. 19-46. 2014
[26] Devaraj Das, Owen O'Malley, Sanjay Radia, and Kan Zhang, "Adding Security to Apache Hadoop", *Hortonworks*, IBM. 2011
[27] Kevin T. Smith, "Big Data Security: The Evolution of Hadoop's Security Model", 2013.
[28] D. J. Bernstein, ''The Salsa20 family of stream ciphers,'' in New Stream Cipher Designs. Berlin, Germany: Springer, pp. 84–97. 3398_3407, Aug.2008
[29] Sumalatha Potteti and Namita Parati, "Secured Data Transfer For Cloud Using Blowfish" *International Journal Of Advances In Computer Science And Cloud Computing*, Issn: 2321-4058 Volume 3, Issue 2, Nov.2015
[30] Papri Ghosh, Vishal Thakor, Dr. Pravin Bhathawala, "Data Security and Privacy in Cloud Computing Using Different Encryption Algorithms" *International Journal of Advanced Research in Computer Science and Software Engineering* 7(5), May, pp. 469-471. 2017
[31] K.Sekar, M Padmavathamma, "Comparative Study of Encryption Algorithm over Big Data in Cloud Systems" 2016 *Internationa conference on Computing for Sustainable Global Development (INDIACom)*.2016
[32] B.Thimma Reddy, K.Bala Chowdappa, S.Raghunath Reddy, "Cloud Security using Blowfish and Key Management Encryption Algorithm" *International Journal of Engineering and Applied Sciences (IJEAS)* ISSN: 2394-3661, Volume 2, Issue 6, June. 2015
[33] Garima Saini and Naveen Sharma, "Triple security of data in cloud computing," *International Journal of Computer Science and Information Technologies*, Vol. 5, No 4, pp.5825-5827, 2014.
[34] Elminaam DS, Abdual-Kader HM, Hadhoud MM,"Evaluating the performance of symmetric encryption algorithms", *IJ Netw Secur* 10:216-222. 2010
[35] Tavares SE, Heys HM, "Avalanche characteristics of substitution-permutation encryption networks." *IEEE Trans Comput* 44:1131-1. 1995.
[36] Shadi Aljawarneh, Muneer Bani Yassein, & We'am Adel Talafha, "A resource-efficient encryption algorithm for multimedia big data," *Multimedia Tools Appl*. 76, 21 (November), 22703-22724. 2017

BIOGRAPHIES OF AUTHORS

Uma Narayanan is a research scholar in the Division of Information Technology, Cochin University of Science and Technology, Kerala, India. She received a bachelor's degree in Computer Science and Engineering and a master's degree in Network Engineering from Mahatma Gandhi University, Kerala, India. She is master in information security, and mainly engages in Big data security research.

Varghese Paul received his B.Sc (Engg) in Electrical Engineering from Kerala University, M.Tech in Electronics and Ph.D. in Computer Science from Cochin University of Science and Technology. Research Supervisor of Cochin University of Science and Technology, M G University Kottayam, Anna Technical University Chennai, Bharathiar University Coimbatore, Bharathidasan University Trichy and Karpagam University Coimbatore. Under the guidance, 29 research scholars had already completed research studies and degree awarded. Research areas are Data Security using Cryptography, Data Compression, Data Mining, Image Processing and E_Governance. Developed TDMRC Coding System for character representation in computer systems and encryption system using this unique coding system. Published many research papers in international as well as national journals and a textbook also.

Shelbi Joseph received the BE. Degree from the University of Madras in 1992 in Computer Science and M.Tech degree in Computer Science from the Department of Computer Science, National Institute of Technology, Tiruchirappalli in 2006. He spent seven years in the software industry, and currently working as Assistant professor, Division of Information Technology, School of Engineering, Cochin University of Science and Technology. He carried out his research work leading to Ph.D. at School of Engineering, Cochin University of Science and Technology in Software Reliability. His areas of interest are Software Engineering, Software Reliability, Open Source Software, Big Data, Data Mining and IOT. He has number of publications in National and International Journals and Conference proceedings to his credit.